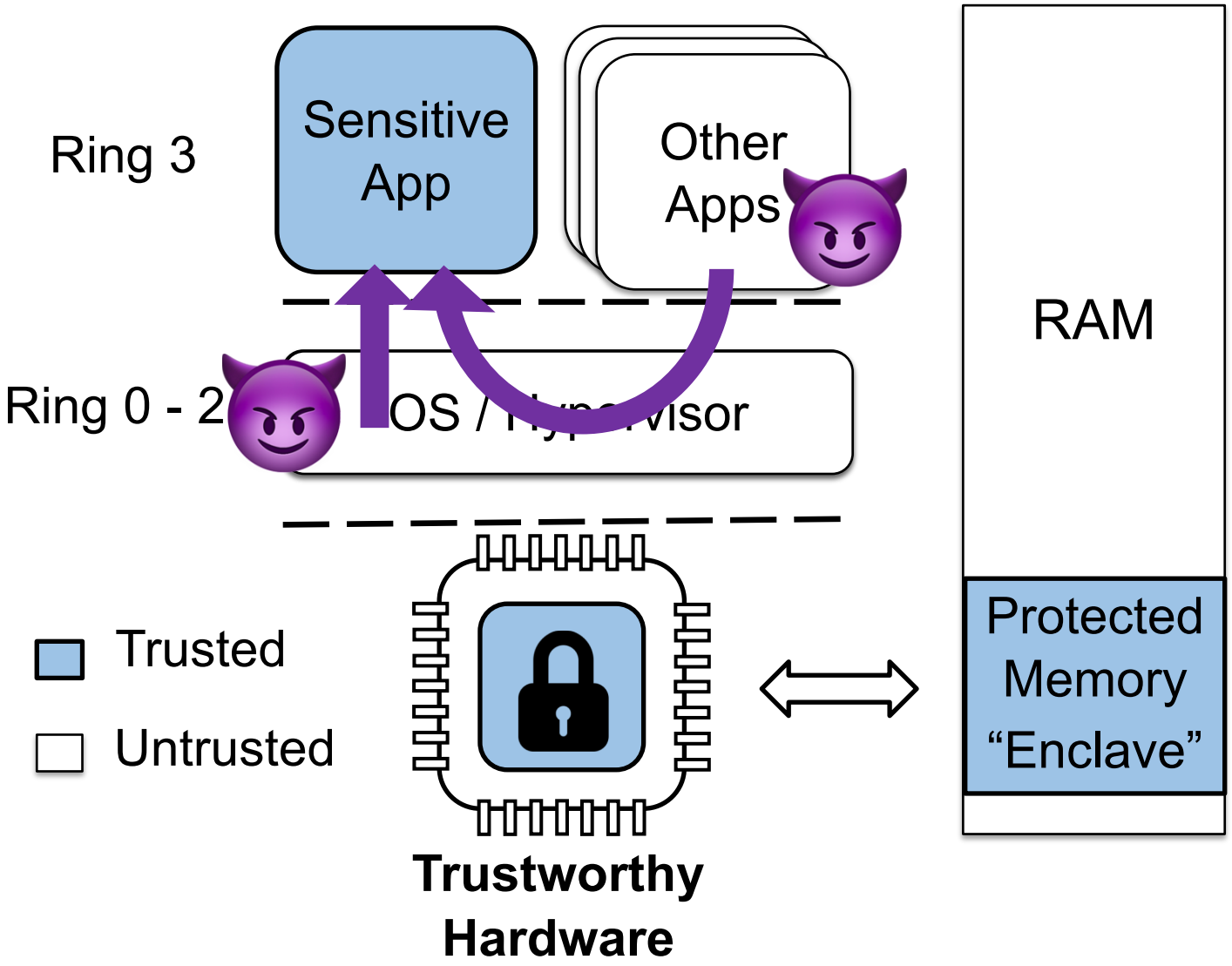
Berkeley
UNIVERSITY OF CALIFORNIA

Keystone: An Open Framework for Architecting Trusted Execution Environments

Dayeol Lee, David Kohlbrenner, Shweta Shinde, Krste Asanovic, Dawn Song

Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley

Trusted Execution Environments (TEEs)

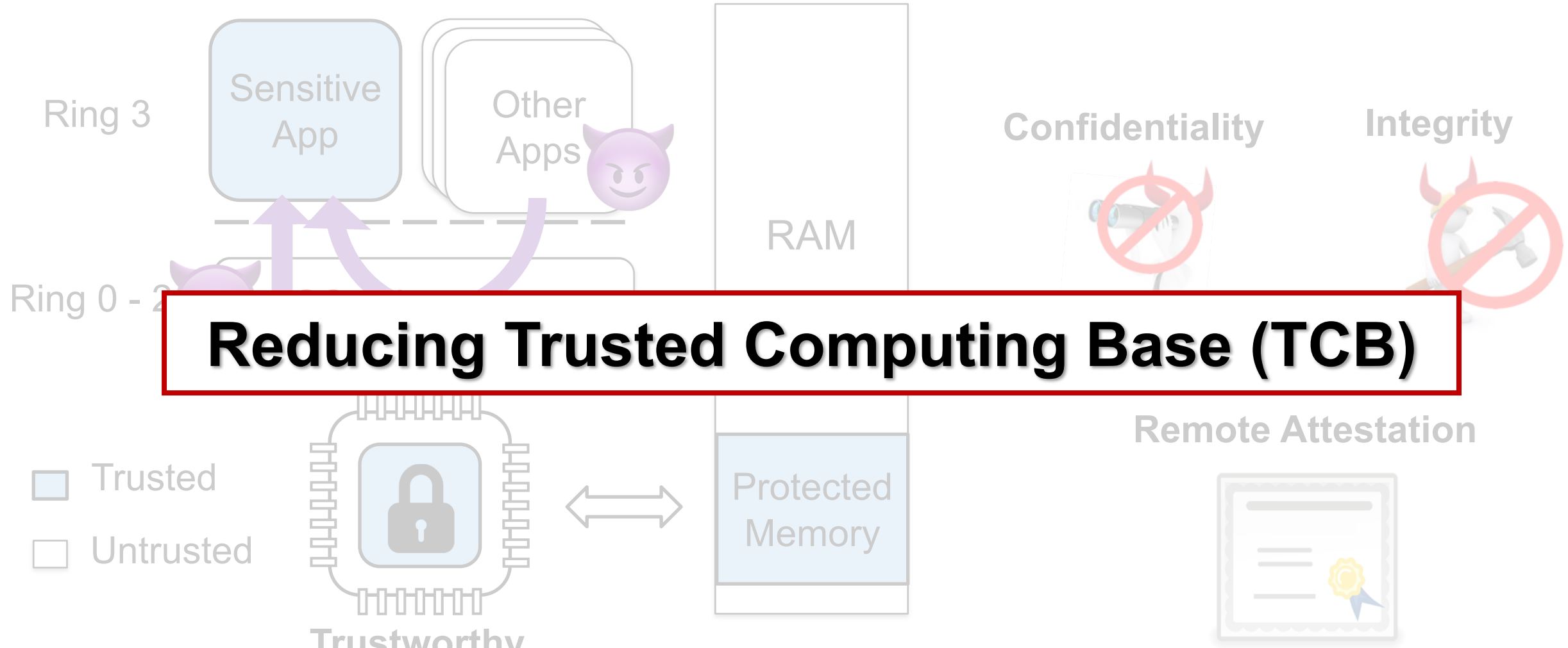


Confidentiality

Integrity

Remote Attestation

Trusted Execution Environments (TEEs)

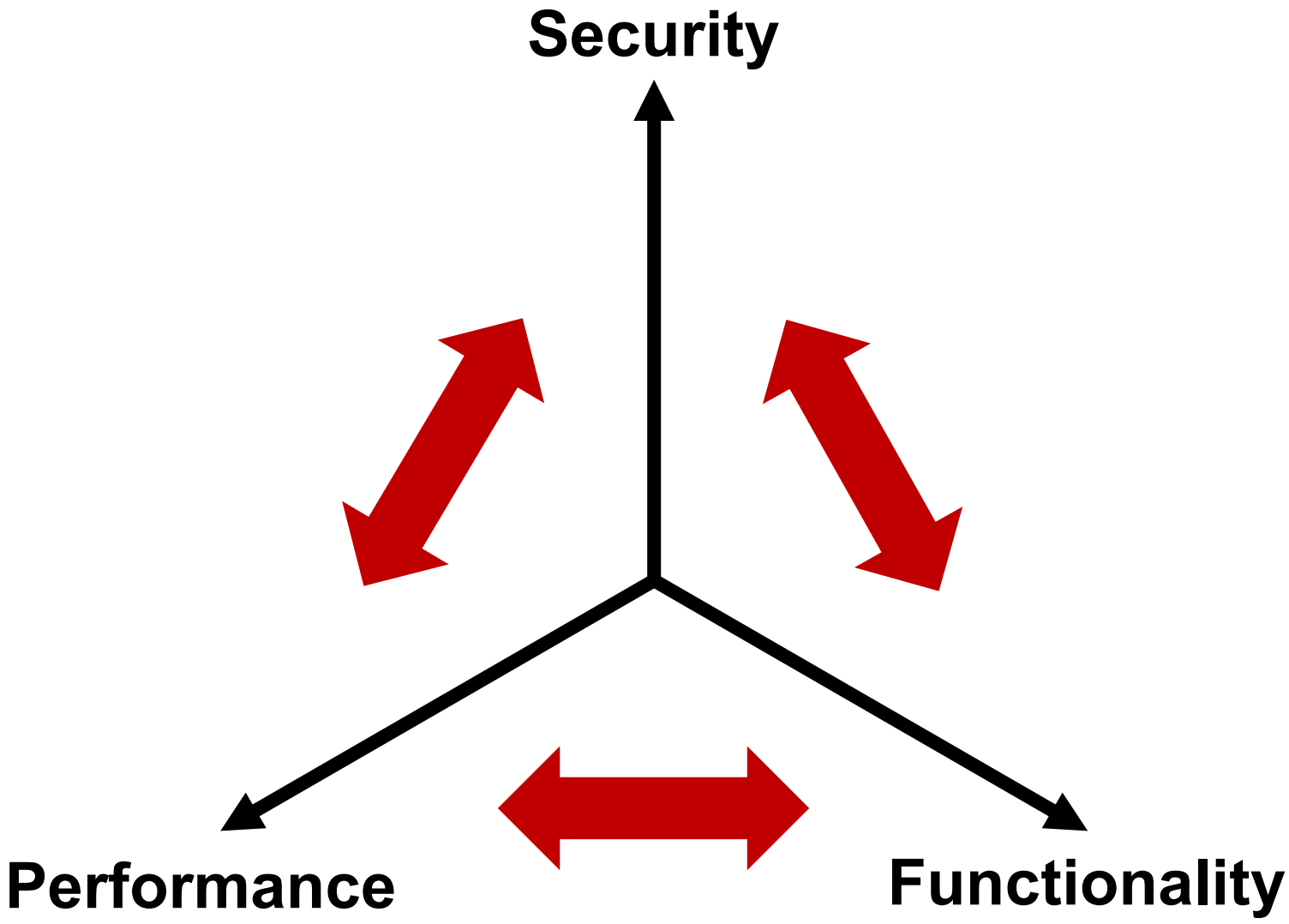


Challenges in Existing TEEs

intel SGX

ARM TrustZone

AMD SEV

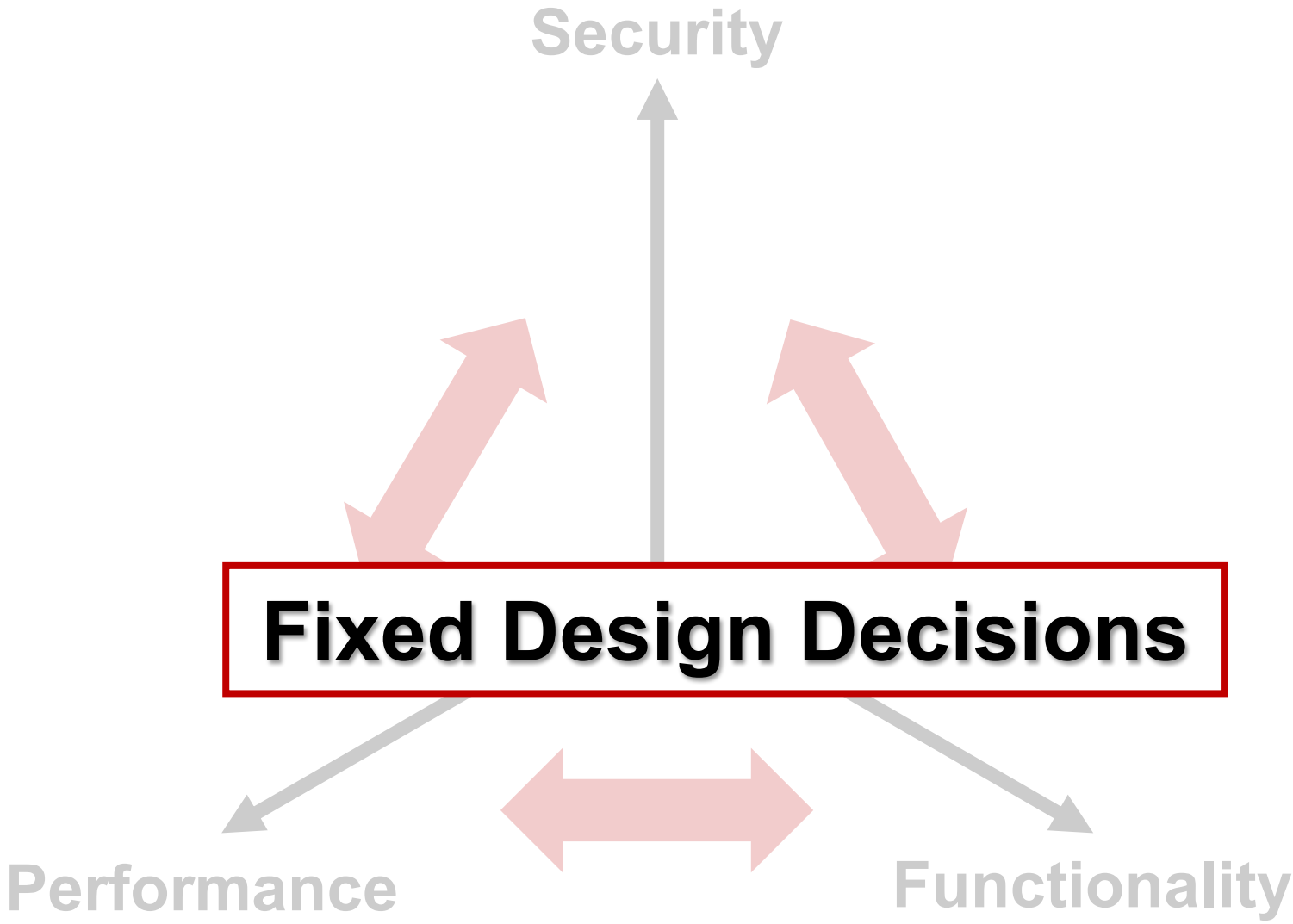


Challenges in Existing TEEs

intel SGX

ARM TrustZone

AMD SEV



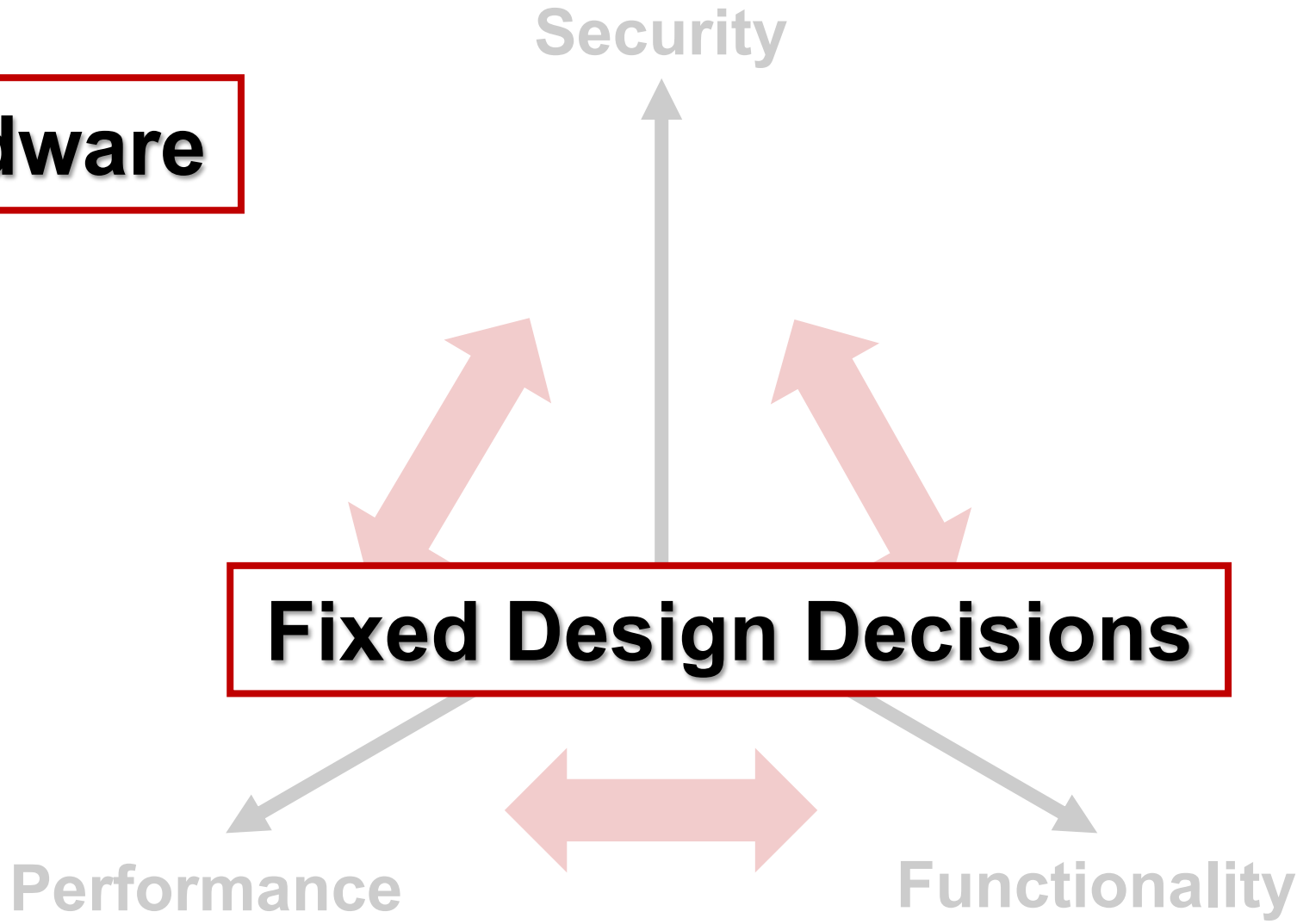
Challenges in Existing TEEs

Closed-Source Hardware

intel SGX

ARM TrustZone

AMD SEV



Technical Contributions

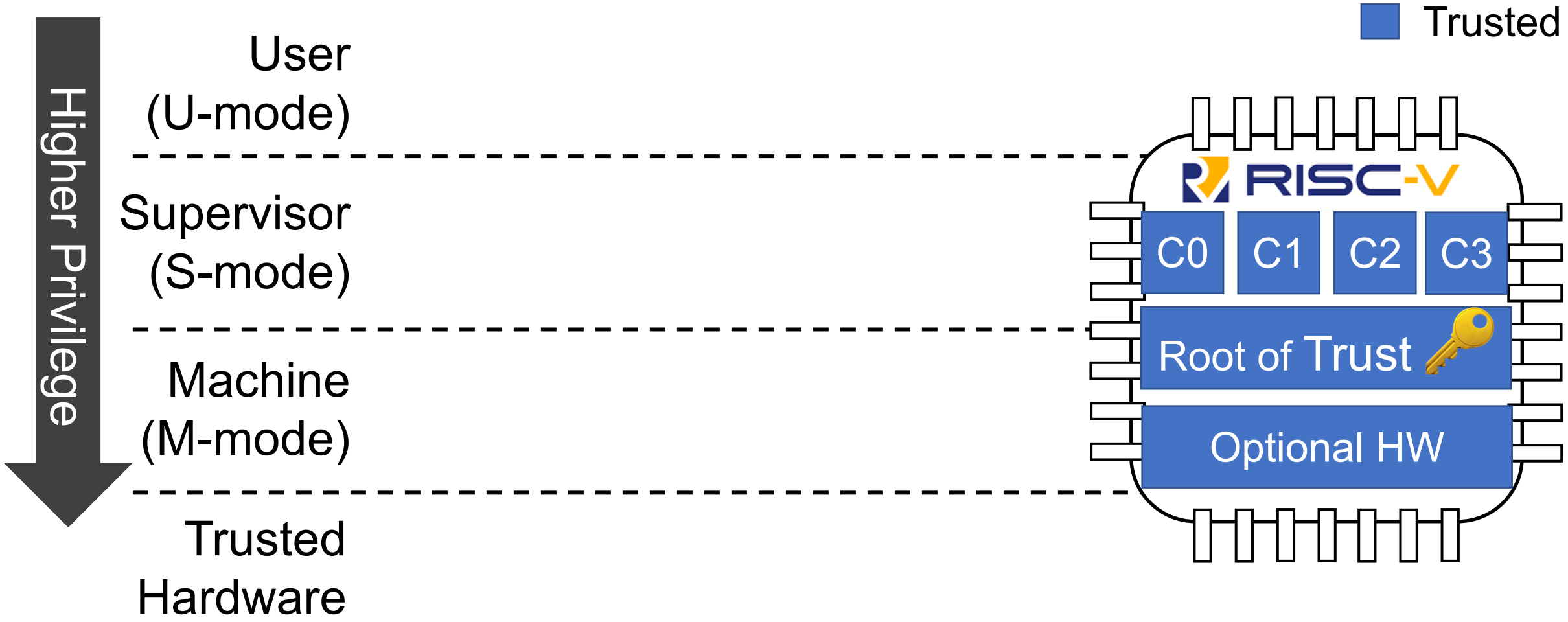
Keystone

- Keystone: Customizable RISC-V TEEs

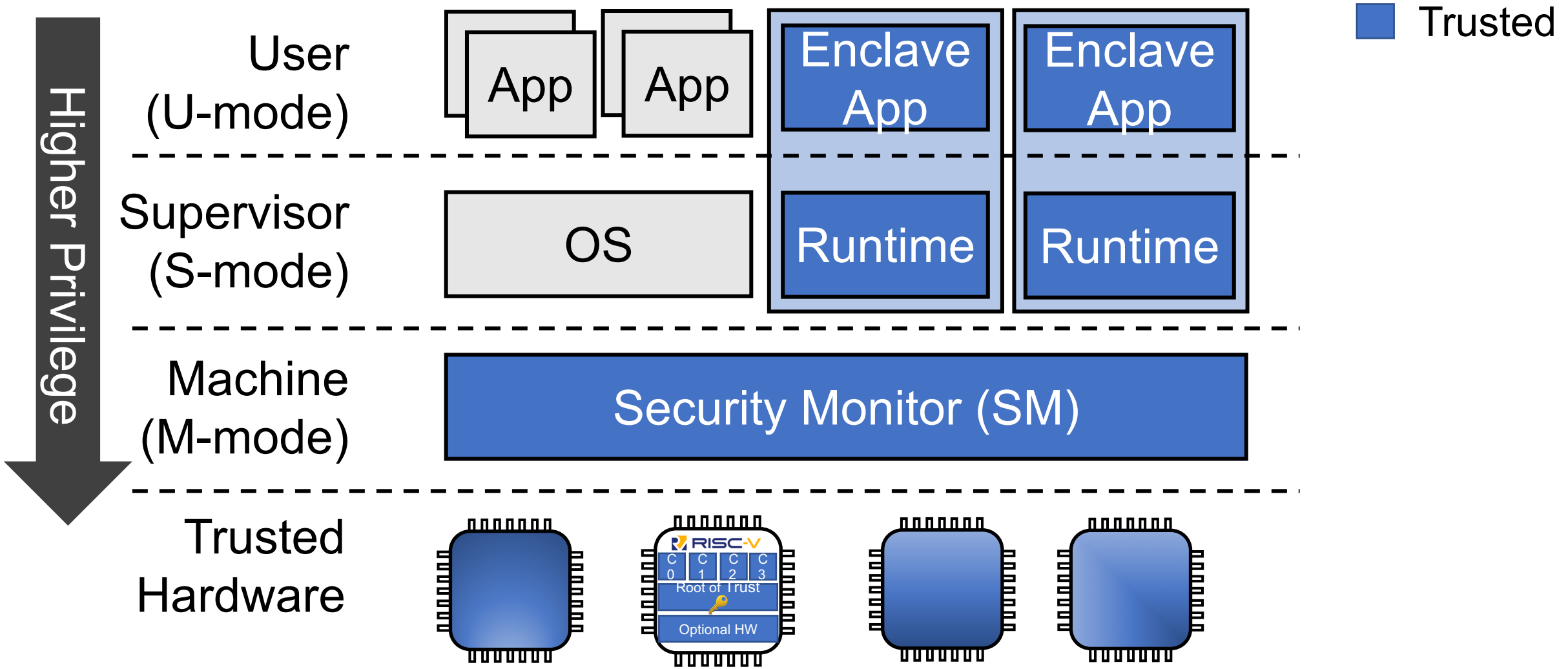
Fine-Grained Configuration	Modular Extensions	Minimal TCB	No μarch Modification

- Framework
  - Extensive benchmarking
  - Real-world applications
  - Multi-platform deployment
- Open-Source
  - Full-stack available
  - Community-driven efforts
  - TEE verification & research

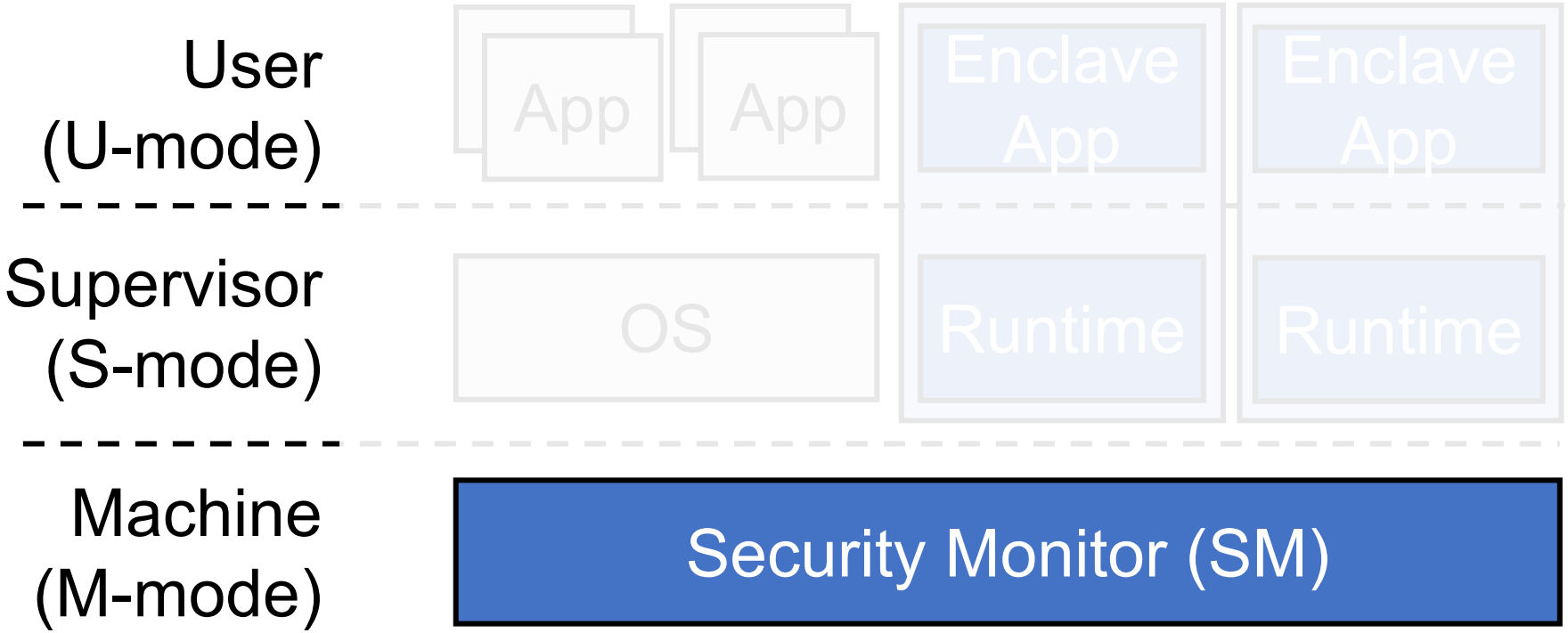
Keystone Architecture and Trust Model



Keystone Architecture and Trust Model

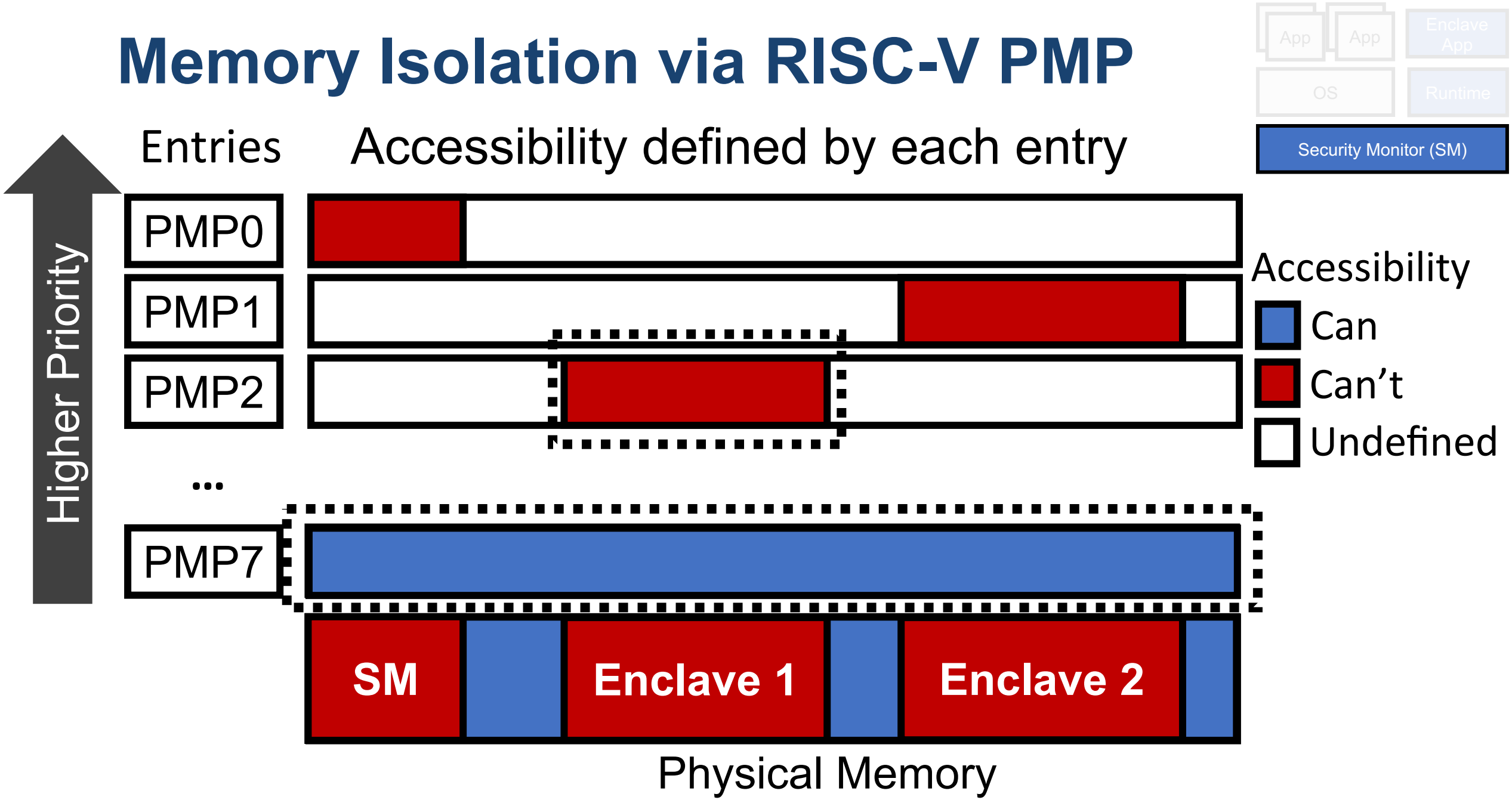


Keystone Architecture and Trust Model

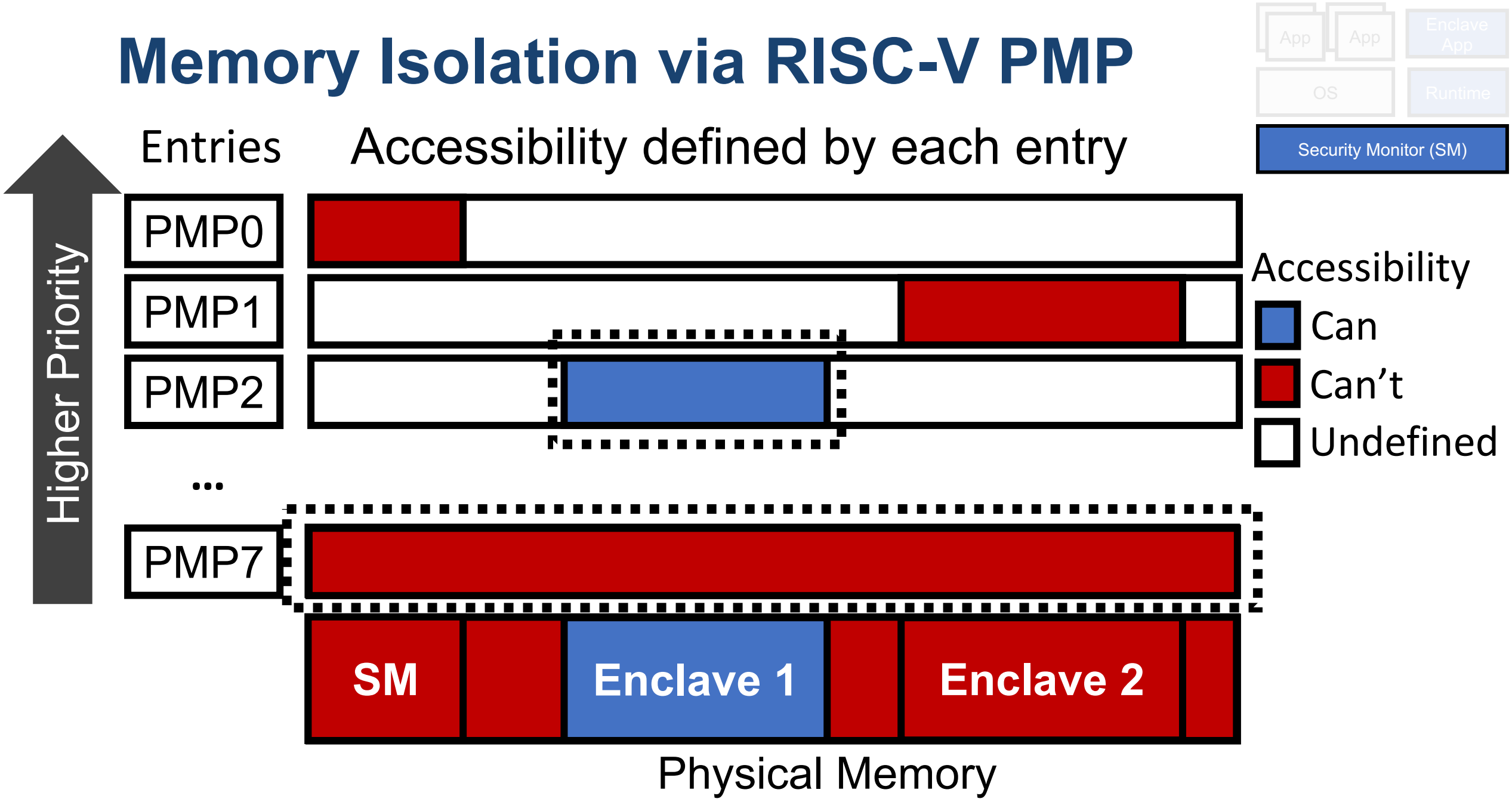


Hardware-Enforced and Software-Defined Isolation

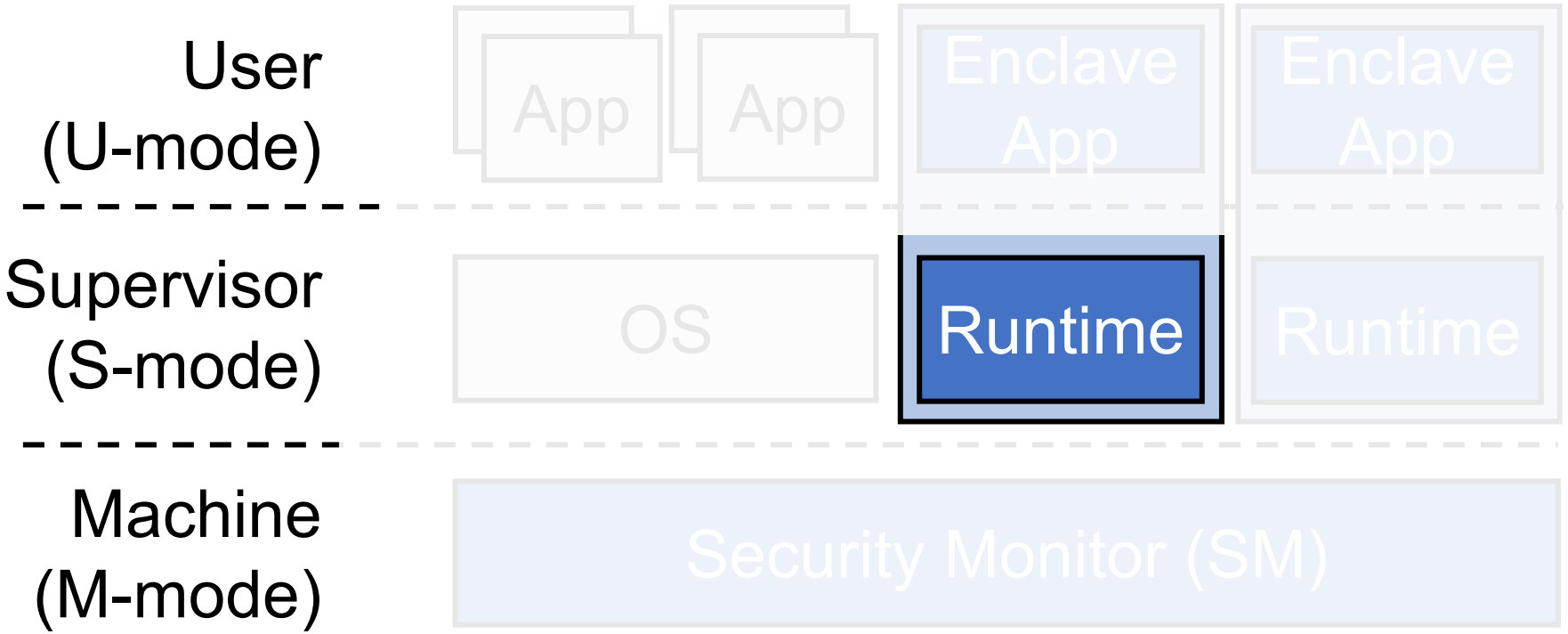
Memory Isolation via RISC-V PMP



Memory Isolation via RISC-V PMP



Keystone Architecture and Trust Model



What Does Keystone Runtime Do?

What does Keystone Runtime Do?

User (U-mode)

App

Enclave App

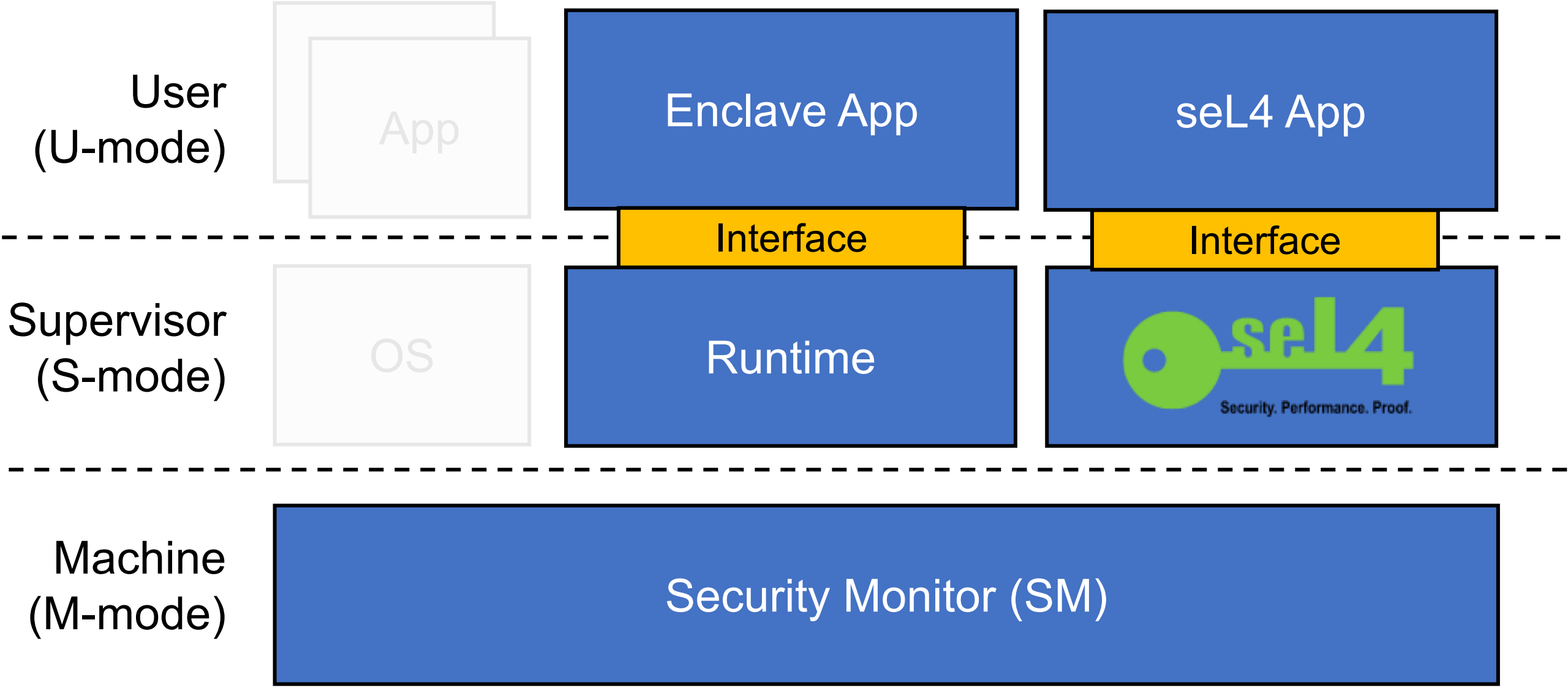
Supervisor (S-mode)

OS

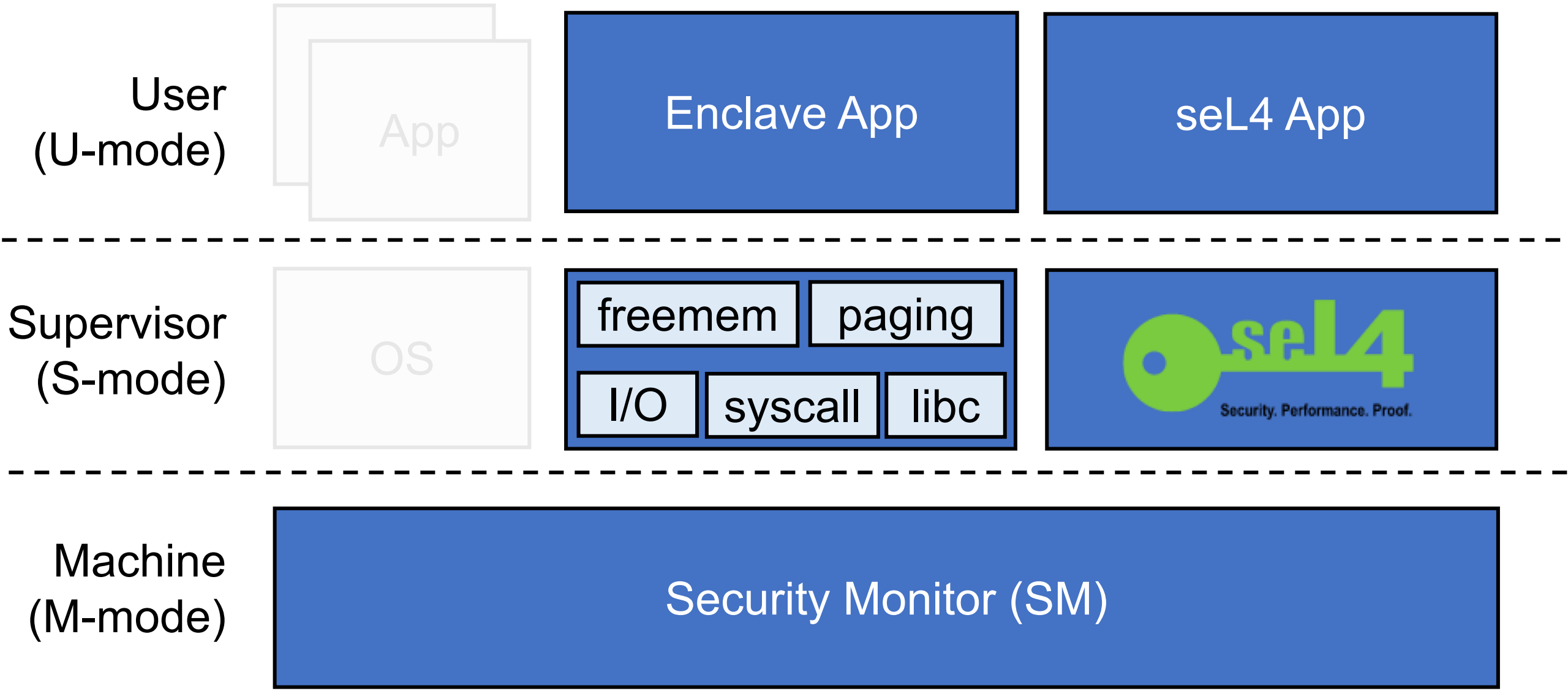
Machine (M-mode)

Security Monitor (SM)

What does Keystone Runtime Do?

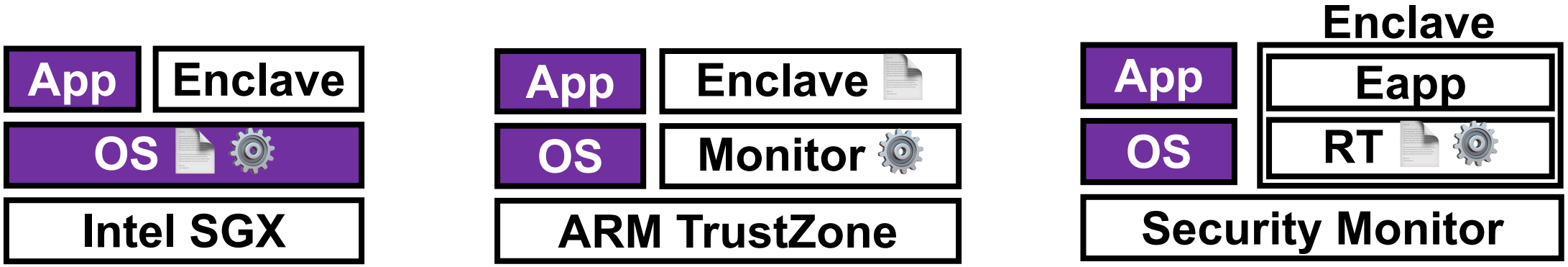


What does Keystone Runtime Do?



Memory Management in Keystone

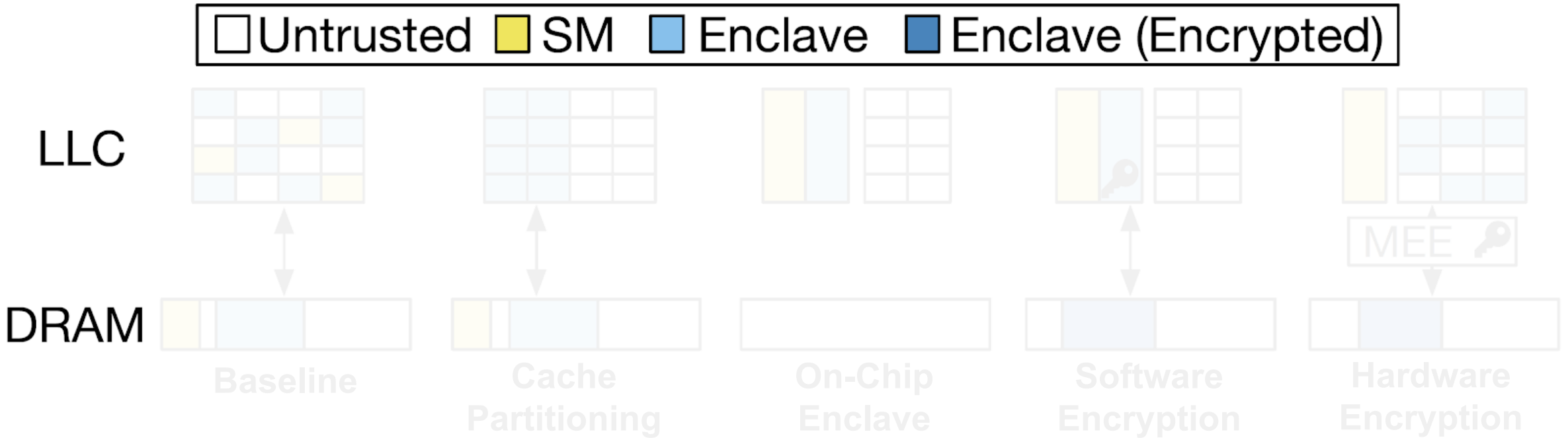
😈 = untrusted 📄 = page table ⚙ = management



Intel SGX **Komodo** **Keystone**

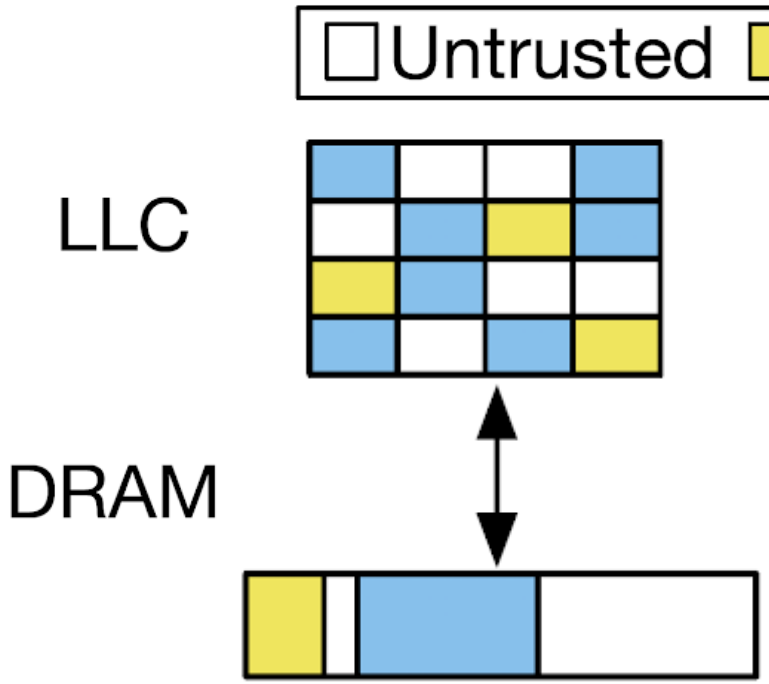
- ❑ Enclave self resource management (e.g., dynamic memory resizing)
- ❑ Various memory protection mechanisms

Various Memory Protection Mechanisms



Adversary					
SW					
Cache SC					
HW					

Various Memory Protection Mechanisms



Baseline

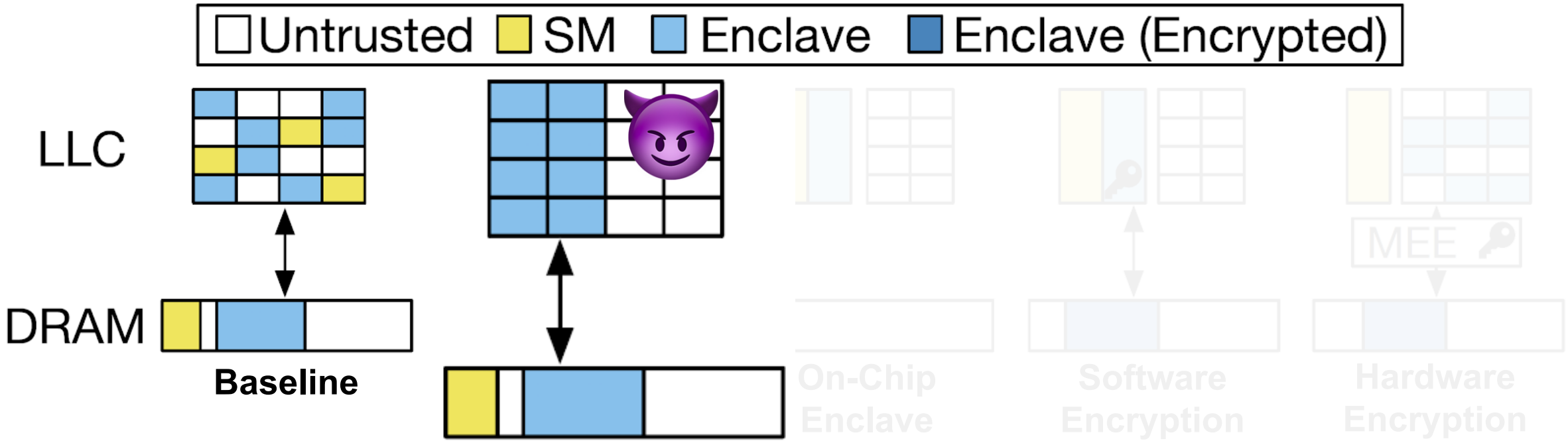
Adversary

SW

Cache SC

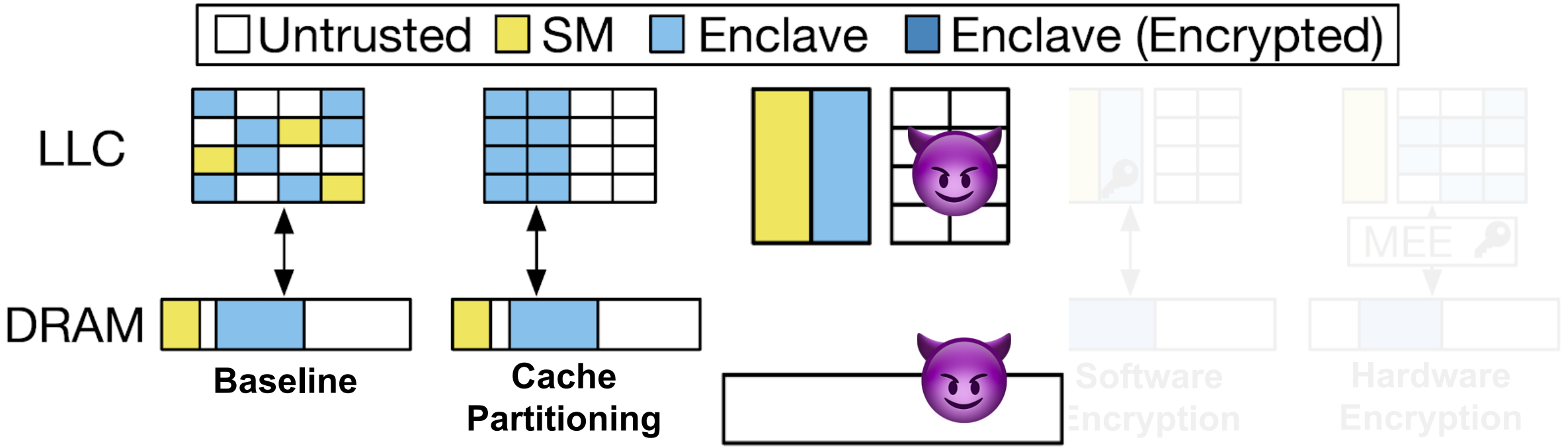
HW

Various Memory Protection Mechanisms



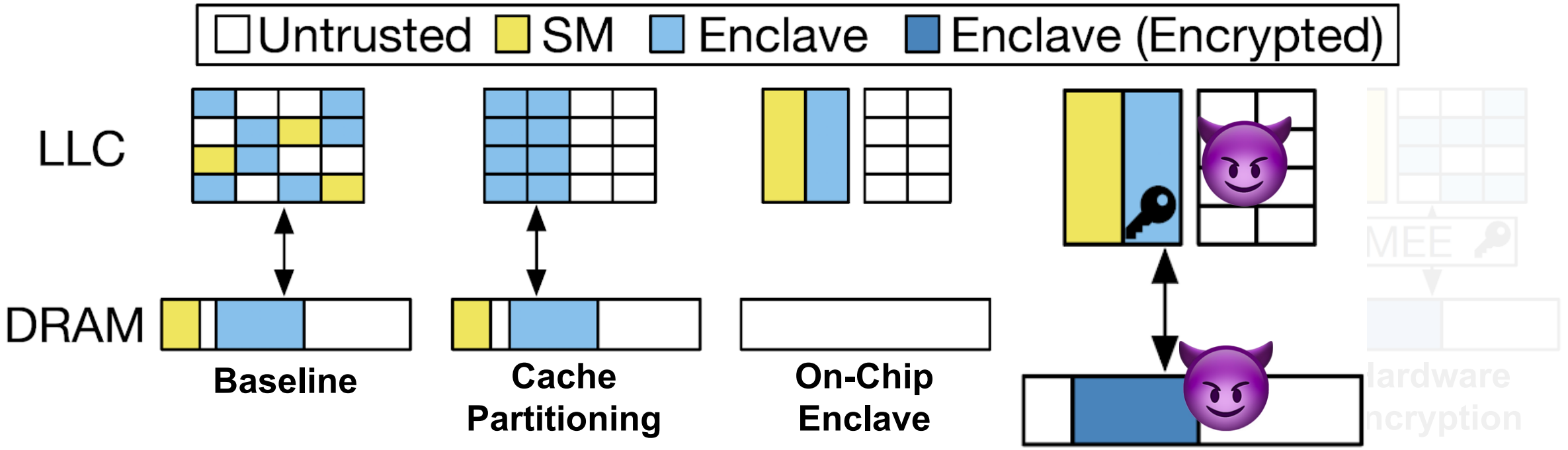
Adversary					
SW	●	●			
Cache SC		●			
HW					

Various Memory Protection Mechanisms



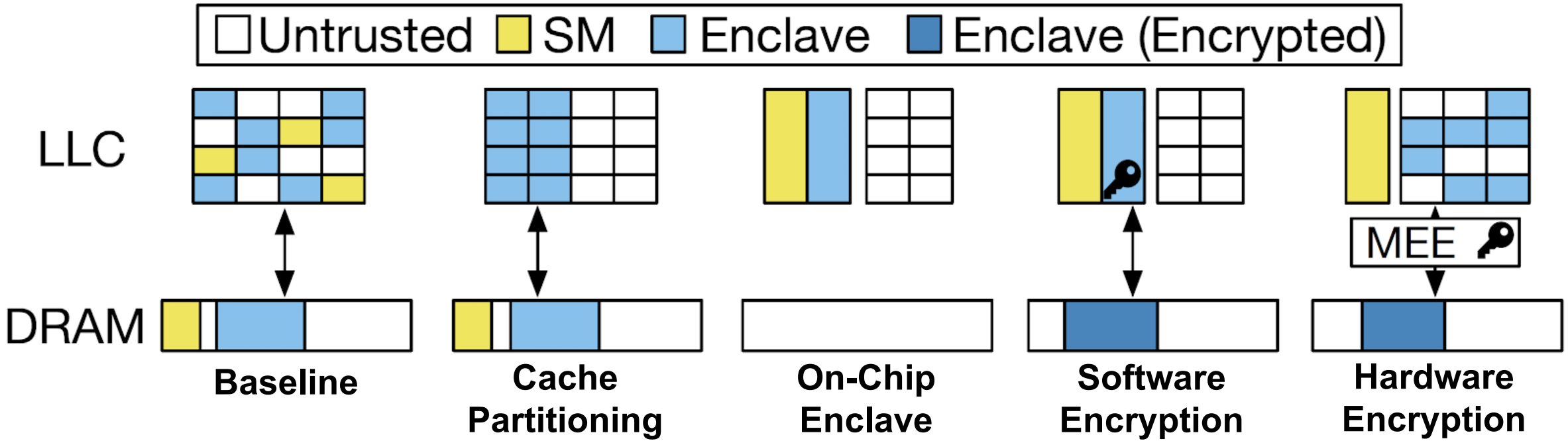
Adversary	Baseline	Cache Partitioning	On-Chip Enclave	Software Encryption	Hardware Encryption
SW	●	●	●		
Cache SC		●	●		
HW			●		

Various Memory Protection Mechanisms



Adversary	Baseline	Cache Partitioning	On-Chip Enclave	Software Encryption	Hardware Encryption
SW	●	●	●	●	●
Cache SC		●	●	●	●
HW			●	●	●

Various Memory Protection Mechanisms



Adversary	Baseline	Cache Partitioning	On-Chip Enclave	Software Encryption	Hardware Encryption
SW	●	●	●	●	●
Cache SC		●	●	●	●
HW			●	●	●

Evaluation

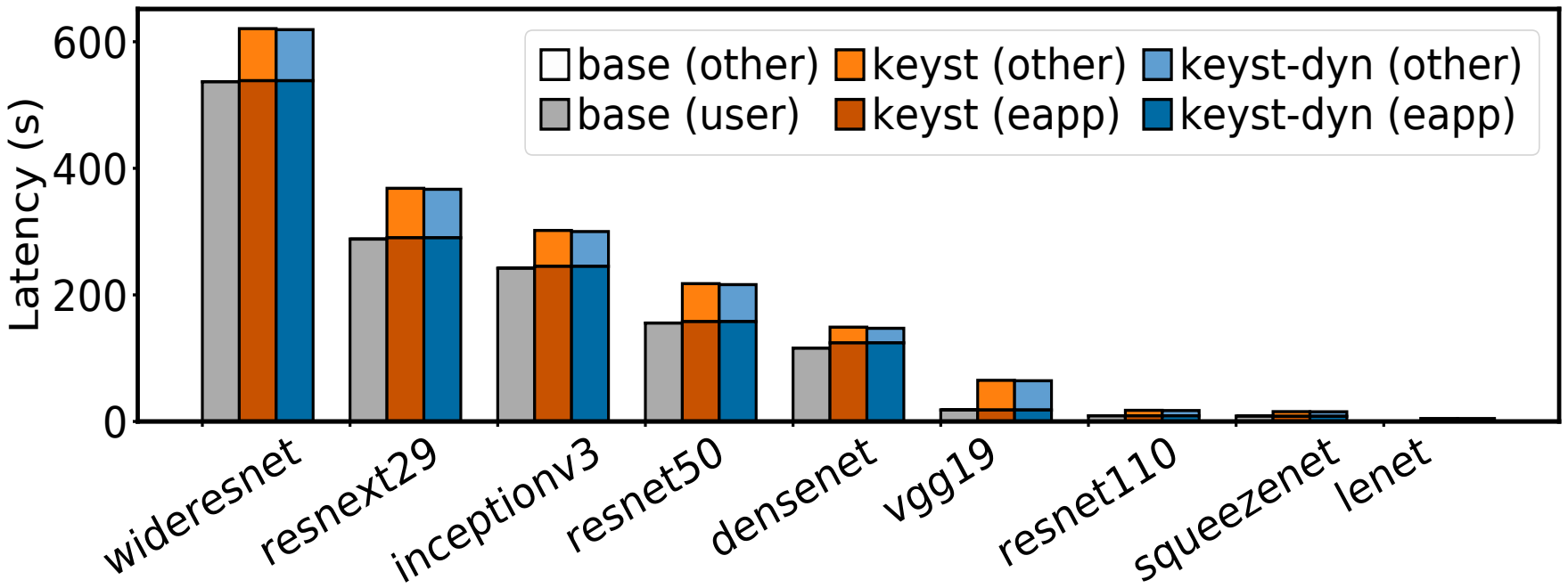
- Security Analysis
  - Keystone enclave defends various adversary models
- Modularity Analysis
  - Keystone supports fine-grained and modular configuration
- Trusted Computing Base Analysis
  - Various of real-world applications with a few thousands of LoC
- Performance Analysis
  - Security Monitor Overhead
  - Runtime Overhead
  - Cost of Memory Protection Mechanisms

Evaluation

- ❑ Security Analysis
  - ➢ Keystone enclave defends various adversary models
- ❑ Modularity
  - ➢ Keystone
- ❑ Trusted Computing Base Analysis
  - ➢ Various of real-world applications with less than thousands of LoC
- ❑ Performance Analysis
  - ➢ Security Monitor Overhead
  - ➢ Runtime Overhead
  - ➢ Cost of Memory Protection Mechanisms

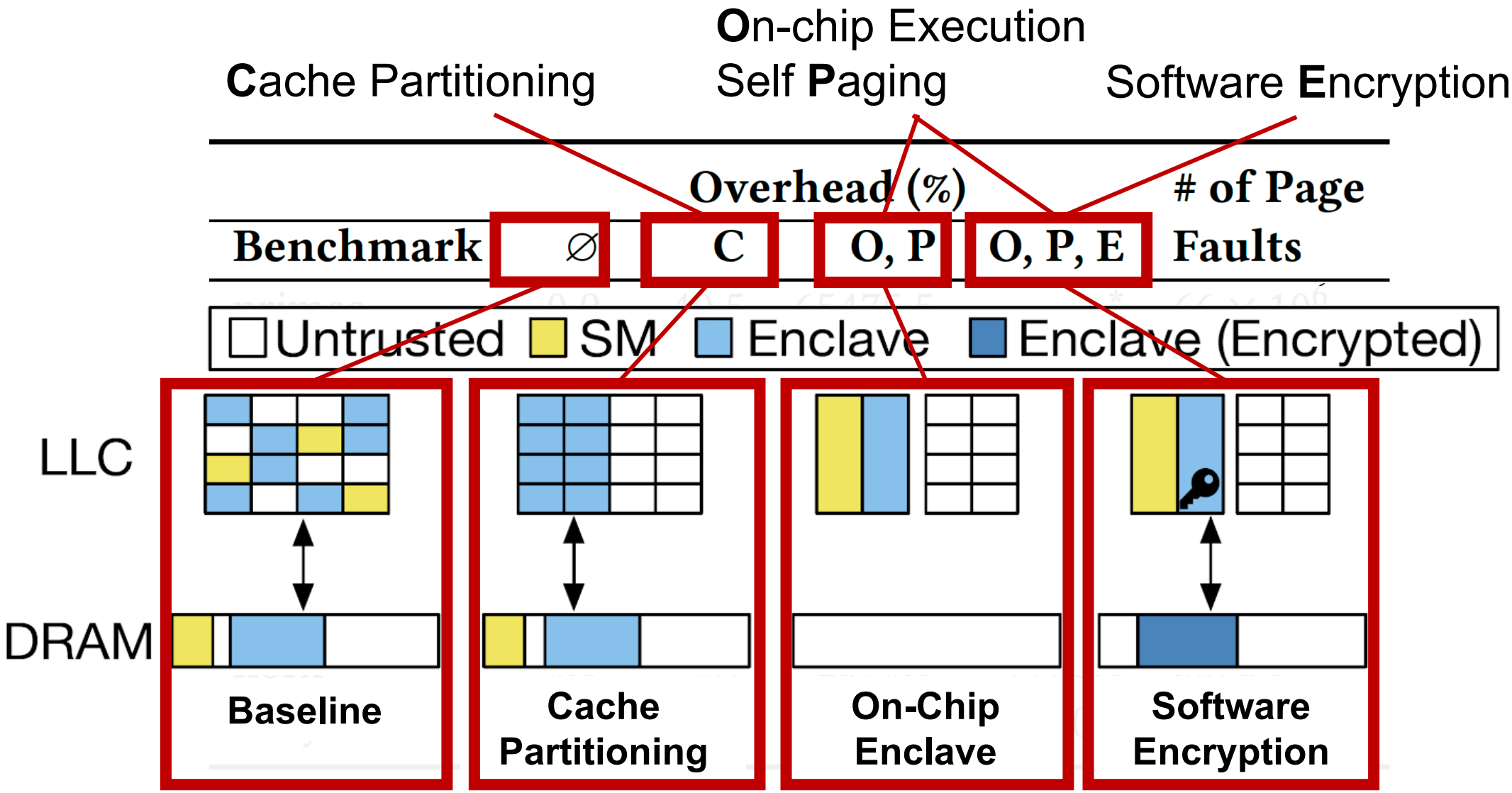
Please check our paper!

Runtime Overhead: Memory Management



- Torch benchmark
  - Unmodified NN inference
- Initialization overhead
  - Enclave measurement (SHA3)
- Execution overhead
  - Min -3.12% (LeNet)
  - Max 7.35% (DenseNet)
- Dynamic memory resizing
  - No noticeable overhead

Cost of Memory Protection Mechanisms



Cost of Memory Protection Mechanisms

Cache Partitioning

On-chip Execution
Self **P**aging

Software **E**ncryption

	Overhead (%)				# of Page
Benchmark	**∅**	**C**	**O, P**	**O, P, E**	**Faults**
primes	-0.9	40.5	65475.5	*	66×10^6
miniz	0.1	128.5	80.2	615.5	18341
aes	-1.1	66.3	1471.0	4552.7	59716
bigint	-0.1	1.6	0.4	12.0	168
qsort	-2.8	-1.3	12446.3	26832.3	285147
sha512	-0.1	0.3	-0.1	-0.2	0
norx	0.1	0.9	2590.1	7966.4	58834
dhrystone	-0.2	0.3	-0.2	0.2	0

Conclusion

- Introduced Keystone, a *customizable* framework for TEEs
- Modular design with fine-grained customizability
- Useful for building TEEs for different threat models, functionality, and performance requirements
- Keystone is fully open-source under BSD 3-Clause
  - https://keystone-enclave.org

Thank You!